

Low-Complexity Hybrid Form FIR Filters Using Matrix Multiple Constant Multiplication

Oscar Gustafsson^a, Jeffrey O. Coleman^b, Andrew G. Dempster^c, and Malcolm D. Macleod^d

^aDepartment of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, oscarg@isy.liu.se

^bNaval Research Laboratory, Radar Division, Washington, DC, USA, jeffc@alum.mit.edu

^cDepartment of Electronic Systems, University of Westminster, London, United Kingdom, dempsta@wmin.ac.uk

^dQinetiQ Ltd., St Andrews Rd, Malvern, WR14 3PS, United Kingdom, mdmacleod@iee.org

Abstract — Hybrid form FIR filters have been shown to provide a trade-off between the direct form and transposed direct form FIR filters resulting in a low power implementation. However, the use of multiple constant multiplication (MCM) techniques is less advantageous as it results in several MCM blocks. In this work a method of implementing low-complexity hybrid form FIR filters using matrix multiple constant multiplication blocks is proposed. The utilized filter structure can be seen as a polyphase decomposition with common delay lines for the subfilters.

I. INTRODUCTION

Finite-length impulse response (FIR) filters are commonly used in many DSP systems. The transfer function of an N th-order FIR filter can be written as

$$H(z) = \sum_{i=0}^N h_i z^{-i} \quad (1)$$

A direct realization of (1) for $N = 5$ is shown in Fig. 1 (a). This filter form is referred to as a direct form FIR filter. If the signal flow graph is transposed the filter structure in Fig. 1 (b) is obtained, referred to as transposed direct form. From an implementation point of view both structures contain the same number of multipliers, adders, and delay elements. However, the width of the delay elements for the transposed direct form is generally larger, depending on where the quantization is performed. Furthermore, the fan-out of the input node is high as one node is driving several multipliers. On the other hand, the critical path for the transposed direct form FIR filter is only one multiplier and one adder compared with one multiplier and $\lceil \log_2(N+1) \rceil$ adders for the direct form (when the additions are rearranged in a binary tree).

A trade-off between the possibly large number of registers for the transposed direct form and the long critical path of the direct form is the hybrid form FIR filter [1]–[4]. Here, any number of delay elements are moved to the opposite branch using retiming, as illustrated in Fig. 1 (c) where the middle delay element of the direct form filter is moved to the opposite branch. It is worth noting that moving all delay elements to the opposite

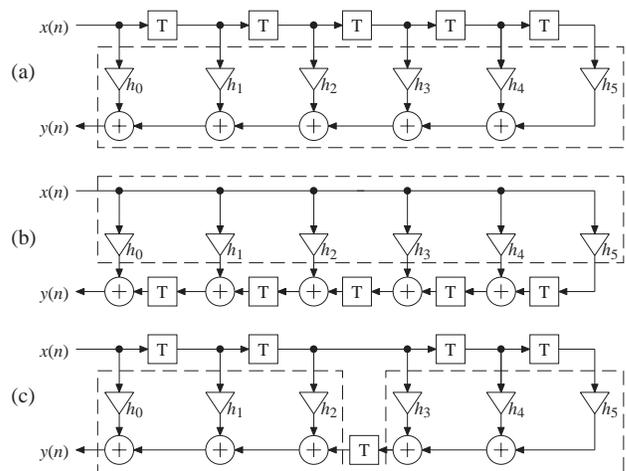


Figure 1. Different realizations of a fifth-order (six tap) FIR filter.

(a) Direct form, (b) transposed direct form, and (c) hybrid form (one of several possible).

branch is identical to transposition. The number of multiplications, additions, and delay elements are kept the same. The critical path is reduced compared to the direct form, while only one delay element have a possibly larger wordlength. The maximum fan-out for a node is also reduced. There are of course several other hybrid forms available providing other trade-offs. It has been noticed that the hybrid form FIR filters are suitable for low-power applications [2],[4].

On the other hand, the implementation of FIR filters using decomposition of the multiplier coefficients into shift-and-add coefficients and utilizing the redundancy between partial results has been an extensively studied area during the last decade [5]–[13]. The aim is to reduce the number of additions/subtractions for the multiplications when one data is multiplied with several constant coefficients, as marked by the dashed box in the transposed direct form FIR filter shown in Fig. 1 (b). The dashed box is referred to as a multiple constant multiplication (MCM) block. Note that transposing the multiplier section leads to a sum-of-products, as marked by the dashed box in the direct form FIR filter shown in Fig. 1 (a). As adders and subtractors have similar implementation costs they will both be referred to as adders.

Several different techniques have been proposed. The proposed algorithms can be divided into three classes. In *subexpression sharing* (or elimination) the idea is to find recurring patterns in the coefficient representation [6]–[8],[10],[13]. The *multiplier block* algorithms uses a constructive algorithm to realize the coefficients from previously realized coefficients [5]. Finally, the *difference methods* finds simple differences between the coefficients to be realized and realizes each coefficients as the sum of another coefficient and a simple difference [9],[11]–[12].

The power consumption of MCM blocks has been studied in [14]. The results give that the number of cascaded adders (adder depth) is a significant measure of power consumption, due to increased glitches in longer combinatorial structures. This would indicate that transposed direct form is more advantageous than direct form considering that aspect.

For the hybrid form FIR filter in Fig. 1 (c) it can be seen that we now have two separate MCM blocks. This leads to that the redundancies can not be utilized to the same degree as for the direct form and transposed direct form FIR filters in Figs. 1 (a) and (b), where all multiplications are in the same MCM block. In this work an FIR filter structure introduced in [15] is utilized. By using this filter structure the previous problem of multiplying one data with several coefficients now is transformed into multiplying a vector with a constant matrix¹. The matrix multiplication can then be simplified using multiple constant techniques for matrices [16]–[19].

A similar structure can be obtained for polyphase decomposed interpolation and decimation FIR filters [20].

In the next section the utilized hybrid form FIR filter form is discussed. Then, in Section III, results are presented where the proposed structure is compared with the traditional hybrid form FIR filter. Finally, some conclusions are given.

II. PROPOSED HYBRID FIR FILTER STRUCTURES

The transfer function for an FIR filter as given in (1) can be divided into subsections of M taps and rewritten, assuming that $N + 1$ is an integer multiple² of M , as

$$H(z) = \sum_{k=0}^{\frac{(N+1)}{M}-1} \left[\sum_{l=0}^{M-1} h_{Mk+l} z^{-l} \right] z^{-Mk} \quad (2)$$

Reordering the sums in (2) gives

-
1. Note that the direct form FIR filter in Fig. 1 (b) can be seen as a scalar product.
 2. If not, the length can be increased by introducing zeros for the extra taps.

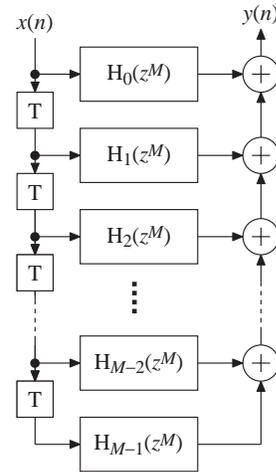


Figure 2. Polyphase decomposed FIR filter.

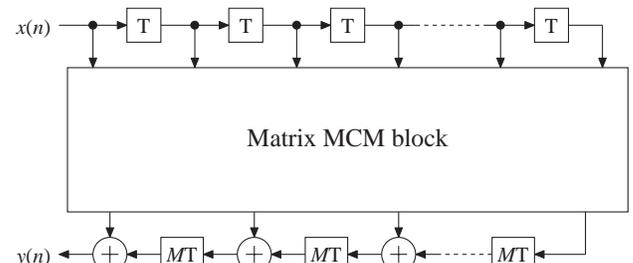


Figure 3. Proposed hybrid form FIR filter with matrix multiple constant multiplication block.

$$\sum_{l=0}^{M-1} \left[\sum_{k=0}^{\frac{(N+1)}{M}-1} h_{Mk+l} z^{-Mk} \right] z^{-l} \quad (3)$$

which is equivalent to the polyphase decomposition of the transfer function. The structure of the polyphase decomposed FIR filter is shown in Fig. 2, where

$$H_l(z) = \sum_{k=0}^{\frac{(N+1)}{M}-1} h_{Mk+l} z^{-Mk} \quad (4)$$

Now, assume that each subfilter in Fig. 2 is composed of a transposed direct form FIR filter. Then, the delay elements of the subfilters can be shared resulting in the structure shown in Fig. 3. Here, the matrix MCM block performs a matrix multiplication as

$$\begin{bmatrix} h_0 & h_1 & \dots & h_{M-1} \\ h_M & h_{M+1} & \dots & h_{2M-1} \\ \dots & \dots & \dots & \dots \\ h_{N+1-M} & h_{N-M+2} & \dots & h_N \end{bmatrix} \quad (5)$$

resulting in the transfer function

$$\begin{bmatrix} 1 \\ z^{-M} \\ \dots \\ z^{-N-1+M} \end{bmatrix}^T \begin{bmatrix} h_0 & h_1 & \dots & h_{M-1} \\ h_M & h_{M+1} & \dots & h_{2M-1} \\ \dots & \dots & \dots & \dots \\ h_{N+1-M} & h_{N-M+2} & \dots & h_N \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \\ \dots \\ z^{-M+1} \end{bmatrix} \quad (6)$$

which is identical to that in (1). Notice that for $M = 1$ the filter structure is identical to the transposed direct form FIR filter and for $M = N + 1$ the filter structure is identical to the direct form FIR filter. The number $M - 1$ can be seen as the number of equally spaced delay elements that has been retimed from the lower (output) branch to the upper (input) branch of the transposed direct form filter in Fig. 1 (b). Then, additional retiming must be applied to end up with the final structure that includes a matrix multiplication.

The filter structure in Fig. 3 have the same number of multipliers, adders, and delay elements as any of the structures in Fig. 1, provided that $N + 1$ is an integer multiple of M . However, in a similar way as for the traditional hybrid form FIR filters, the critical path is reduced compared to the direct form filter and the number of wide delay elements are smaller than for the transposed form FIR filter. Furthermore, as opposed to the traditional hybrid form FIR filter, we no longer have a number of smaller MCM blocks, but one matrix MCM block. This not only allows us to utilize redundancies between the coefficients in a row or a column, but also between complete rows or columns.

If the signal flow graph in Fig. 3 is transposed the resulting transfer function becomes

$$\begin{bmatrix} 1 \\ z^{-1} \\ \dots \\ z^{-M+1} \end{bmatrix}^T \begin{bmatrix} h_0 & h_M & \dots & h_{N+1-M} \\ h_1 & h_{M+1} & \dots & h_{N-M+2} \\ \dots & \dots & \dots & \dots \\ h_{M-1} & h_{2M-1} & \dots & h_N \end{bmatrix} \begin{bmatrix} 1 \\ z^{-M} \\ \dots \\ z^{-N-1+M} \end{bmatrix} \quad (7)$$

resulting in a different matrix multiplication and with different number of delay elements in the top and bottom row of Fig. 3. However, as the matrix in (7) is the transpose of the matrix in (6), they require the same number of additions, except for the possible change of dimension.

III. RESULTS

A 62nd-order FIR filter designed in [21] is used as an example. The matrix MCM blocks are designed using the algorithm in [19], while the MCM blocks for the reference hybrid filters are designed using the algorithm in [5]. The algorithm in [5] is known to produce the best results published for the MCM problem.

In Fig. 4 the number of adders for varying M is shown. For the traditional hybrid form $M - 1$ denotes the number of delay elements in the input delay line, similarly as discussed for the

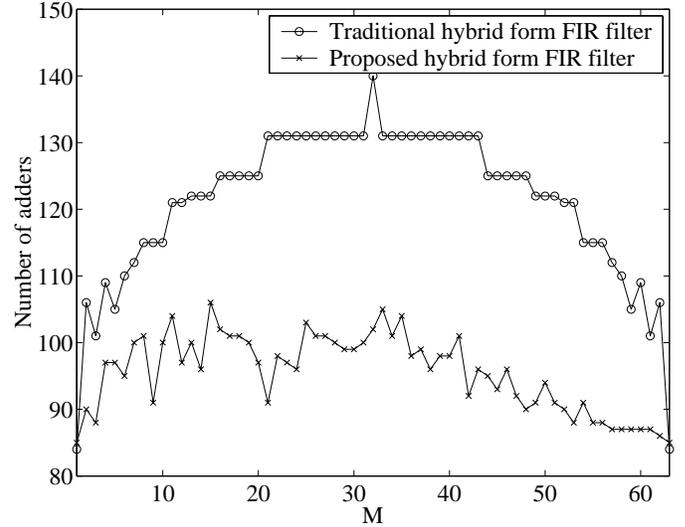


Figure 4. Resulting number of adders for the complete filter in the results section using the proposed hybrid FIR filter form compared with the traditional hybrid FIR filter form.

proposed filter structure. Hence, $N - M + 1$ is the number of delay elements in the output delay line. This leads to that the traditional hybrid form filter will have a symmetrical number of adders as the coefficients in the resulting MCM blocks for $M = K$ and $M = N - K + 1$ are identical. Notice that the number of delay elements for the introduced filter structure is $\lceil (N + 1) / M \rceil M - 1$. Hence, for certain M , typically large M , the resulting number of delay elements may be significantly larger.

From Fig. 4, it is clear that utilizing the proposed hybrid form FIR filter reduces the number of adders compared with the traditional hybrid form. The fact that the traditional form requires fewer adders for $M = 1$ and 63 is due to the algorithms [5] producing relatively better results than the algorithm in [21]. An optimal matrix MCM algorithm would always produce at least as good results.

Furthermore, it can also be seen from Fig. 4 that the worst case for the traditional hybrid form structure is when every other delay element is retimed to the opposite branch ($M = 32$). For this case each MCM block consists of at most two multiplications. However, for the introduced hybrid form filter this situation is not a problem.

An additional degree of freedom for the hybrid form FIR filter is the possibility to distribute the extra zeros required when M is not an integer multiple of $N + 1$. These can be placed either at the start or the end of the impulse response and yield different matrices. For most cases it is simple to just try all different combinations. Clear variations have been observed.

Different matrix MCM algorithms have different properties. The algorithm in [18] and to some extents the algorithm in [17] works better for matrices with many rows and few columns as the methods of sharing are more row oriented. The algorithms in [16] and [19] is based on pattern matching, and, hence, should give results whose quality is independent of the matrix.

IV. CONCLUSIONS

In this work a hybrid form FIR filter structure that allows the use of matrix multiple constant multiplication (MCM) blocks were considered. The considered hybrid form FIR filter can also be seen as a special case of a polyphase decomposed filter with common delay elements for the subfilters. It was shown by an example that the considered structure requires fewer adders than the traditional hybrid form filter. The benefits of hybrid form FIR filters over direct and transposed direct form filters are that the critical path, maximum fan-out for a node, and the number of wider delay elements can be traded for each other, possibly reducing the power consumption. By utilizing the considered structure these benefits are obtained, while still keeping the possibility to significantly reduce the number of additions/subtractions using MCM techniques.

Further evaluation on the circuit level must be performed, but it seems like the proposed structure can be an attractive alternative as both the trade-off possibilities of the hybrid form filters as well as the complexity reduction of MCM are obtained at the same time. This is something that previously have not been able to combine.

REFERENCES

- [1] H.-R. Lee, C.-W. Jen, and C. M. Liu, "A new hardware-efficient architecture for programmable FIR filters," *IEEE Trans. Circuits Syst.-II*, vol. 43, no. 9, pp. 637–644, Sept. 1996.
- [2] K. Azadet and C. J. Nicole, "Low-power equalizer architectures for high speed modems," *IEEE Communications Magazine*, vol. 36, no. 10, pp. 118–126, Oct. 1998.
- [3] K.-Y. Khoo, Z. Yu, and A. N. Willson Jr, "Design of optimal hybrid form FIR filter," in *Proc. IEEE Int. Symp. Circuits Syst.*, Sydney, Australia, May 6–9, 2001, vol. 2, pp. 621–624.
- [4] S.-F. Lin, S.-C. Huang, F.-S. Yang, C.-W. Ku, and L.-G. Chenm "Power-efficient FIR filter architecture design for wireless embedded system," *IEEE Trans. Circuits Syst.-II*, vol. 51, no. 1, pp. 21–25, Jan. 2004.
- [5] A. G. Dempster and M. D. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst.-II*, vol. 42, no. 9, pp. 569–577, Sept. 1995.
- [6] M. Potkonjak, M. B. Srivastava, and A. P. Chandrakasan, "Multiple constant multiplications: efficient and versatile framework and algorithms for exploring common subexpression elimination," *IEEE Trans. Computer-Aided Design*, vol. 15, no. 2, pp. 151–165, Feb. 1996.
- [7] R. I. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," *IEEE Trans. Circuits Syst.-II*, vol. 43, pp. 677–688, Oct. 1996.
- [8] R. Pasko, P. Schaumont, V. Derudder, S. Vernalde, and D. Durackova, "A new algorithm for elimination of common subexpressions," *IEEE Trans. Computer-Aided Design*, vol. 18, no. 1, pp. 58–68, Jan. 1999.
- [9] K. Muhammad and K. Roy, "A graph theoretic approach for synthesizing very low-complexity high-speed digital filters," *IEEE Trans. Computer-Aided Design*, vol. 21, no. 2, Feb. 2002.
- [10] M. Martínez-Peiró, E. Boemo, and L. Wanhammar, "Design of high speed multiplierless filters using a nonrecursive signed common subexpression algorithm," *IEEE Trans. Circuits Syst.-II*, vol. 49, no. 3, pp. 196–203, Mar. 2002.
- [11] O. Gustafsson and L. Wanhammar, "A novel approach to multiple constant multiplication using minimum spanning trees," in *Proc. IEEE Midwest Symp. Circuits Syst.*, Tulsa, OK, Aug. 4–7, 2002, vol. 3, pp. 652–655.
- [12] H. Ohlsson, O. Gustafsson, and L. Wanhammar, "Implementation of low-complexity FIR filters using a minimum spanning tree," in *Proc. IEEE Mediterranean Electrotechnical Conf.*, Dubrovnik, Croatia, May 12–15, 2004, pp. 261–264.
- [13] F. Xu, C.-H. Chang, and C.-C. Jong, "Efficient algorithms for common subexpression elimination in digital filter design," in *Proc. IEEE Int. Conf. Acoustics Speech, Signal Processing*, 17–21 May, 2004, vol. 5, pp. 137–140.
- [14] S. S. Demirsoy, A. G. Dempster, and I. Kale, "Power analysis of multiplier blocks," in *Proc. IEEE Int. Symp. Circuits Syst.*, 26–29 May, 2002, vol. 1, pp. 297–300.
- [15] J. O. Coleman, "A double binary-tree FIR filter form and the corresponding low-sensitivity IIR bandpass/bandstop structure," in *Proc. IEEE Int. Midwest Symp. Circuits Syst.*, Tulsa, OK, Aug. 4–7, 2002.
- [16] N. Boullis and A. Tisserand, "Some optimizations of hardware multiplication by constant matrices," in *Proc. IEEE Symp. Computer Arithmetic*, Santiago de Compostela, Spain, June 15–18, 2003, pp. 20–27.
- [17] A. G. Dempster, O. Gustafsson, and J. O. Coleman, "Towards an algorithm for matrix multiplier blocks," in *Proc. European Conf. Circuit Theory Design*, Kraków, Poland, Sept. 1–4, 2003.
- [18] O. Gustafsson, H. Ohlsson, and L. Wanhammar, "Low-complexity constant coefficient matrix multiplication using a minimum spanning tree approach," in *Proc. Nordic Signal Processing Symp.*, Espoo, Finland, June 9–11, 2004.
- [19] M. D. Macleod and A. G. Dempster, "A common subexpression elimination algorithm for low-cost multiplierless implementation of matrix multipliers," *Electronics Letters*, vol. 40, no. 11, pp. 651–652, 2004.
- [20] O. Gustafsson and A. G. Dempster, "On the use of multiple constant multiplication in polyphase FIR filters and filter banks," in *Proc. Nordic Signal Processing Symp.*, Espoo, Finland, June 9–11, 2004, pp. 53–56.
- [21] Y.-C. Lim and S. R. Parker, "Discrete coefficient FIR filter design based upon an LMS criteria," *IEEE Trans. Circuits Syst.*, vol. 30, pp. 723–739, Oct. 1983.